DÉROULEMENT DE L'ATELIER DÉCOUVERTE D'OUTILS DE SOUS-TITRAGE COLLABORATIFS

Point 0 : Ressources de départ

Disponibilité du film > On part de versions en ligne, l'atelier se placera sur la thématique « sous-titrer une vidéo disponible sur Youtube »
https://www.youtube.com/watch?v=lSqurTMe7Rw
https://www.youtube.com/watch?v=FMvqdaE8S8g
http://www.dailymotion.com/video/xurmcr_we-are-legion-the-story-of-the-hacktivists_news
http://vimeo.com/67919102
https://archive.org/details/WeAreLegionLeaked

Script en anglais avec une timeline pré-existante
>~14000 mots à découper pour l'atelier
Fichier source des sous-titres : We.Are.Legion.2012.DVDRip.XviD-DOSE.srt
http://subtitlesbank.com/fr/we-are-legion-the-story-of-the-hacktivists-english-srt-3668788/

Phase 1 [en amont] : Répartition collaborative des traductions et relectures

Pour des questions de temps, on a fait l'impasse sur des outils de traduction assistée par ordinateurs (TAO) bien qu'il existe des versions libristes telles que OMEGA-T. Cela fera l'objet d'ateliers ultérieurs.

Les traductions ont été réparties de la manière qui suit :

> 8000 mots pris en charge par 4 étudiants de l' UE5 du Master Rédacteurs/Traducteurs de l'UBO dans le cadre des Agences de Traduction Fictives, projet individuel des élèves en M2 via la plateforme collaborative des traducteurs M2

> 6000 mots ont été répartis sur 6-7 pads hébergés par Framasoft et traduits en mode collaboratif par 5 traducteurs volontaires.

Des premières relectures ont été effectuées par 4 contributeurs avec tests des possibilités offertes mais aussi des limites présentées par les mode framapads ou framacalcs. Cela donnera lieu à des réflexions ultérieures pour tenter d'améliorer les flux de travaux répartis en mode collaboratif et illustre la nécessité de prévoir divers types d'instructions et précautions à transmettre aux traducteurs et relecteurs avant le lancement des projets de trads.

-règles typographiques usuelles & de sous-titrages de documentaire
http://lingalog.net/dokuwiki/_media/cours/sg/trad/methodest.pdf

-glossaires
-annotations
-discussions IRC
-synchronisation en temps réel du travail de plusieurs collaborateurs travaillant en simultanée
-taille des textes supportés par les pads
-coexistence du français / anglais plus pratique pour les traducteurs & relecteurs mais galère pour la centralisation. Envisager d'intégrer des caractères spéciaux de type # pour valider les traductions avec un séparateur qui servira à donner des instructions dans d'autres logiciels (par exemple commandes de tri sélectif string commençant par # puis supprimer #) ?

Phase 2 [en amont] : Compilation d'un fichier intégral de sous-titres traduits

Les différents fragments de traduction on été recompilés ensemble dans un fichier final unique qui servira de base à l'atelier de travail effectif sur le sous-titrage collaboratif. S'est alors posée la question des formats de fichier et encodages à respecter pour optimiser le temps de préparation des supports en vue des ateliers.

Fichier exporté en .txt > fonctionne tout simplement avec VLC à la condition que le titre soit rigoureusement identique et le fichier présent dans le même répertoire que la vidéo. Mais le .txt n'est à priori pas pris en compte par des plateformes autres.

Fichier exporté en .srt > fonctionne à priori le mieux avec les différent outils envisagés, notamment grâce à la prise en compte automatique du séparateur temporel « --> ».
Autres formats à tester ultérieurement ?

L'encodage privilégié est l'UTF-8, surtout pas l'ANSI, afin d'éviter les sauts de caractères spéciaux et autres mauvaises surprises. En cours du projet, nous nous sommes heurtés à des problèmes récurrents lors des tentatives de copier/coller simples entre Framapads, Libre Office (odt ou ods) et génération de fichiers .txt ou .srt via Notepad++ ou autres éditeurs de texte. Par défaut et non pas seulement par paresse, nous avons du passer par Excel (:oS) pour simplifier les manip dans un premier temps car il convertissait automatiquement les formats sans rien faire sauter.

Cependant, le fichier unique final a pu être exporté du format .xls en .ods sans perte de travail et sans sauts de caractères. Du fichier .ods on a pu copier/coller les textes compilés sur un nouveau fichier Notepad++, enregistré sous le format .txt (en vérifiant l'encodage). Ce dernier fichier .txt a été transformé en fichier .srt par une modification de l'extension du fichier en mode manuel.

Il faudra explorer les raisons pour lesquelles l 'ods faisait sauter les copier/coller depuis les pads et calcs ?
Taille du fichier peut être ?
Prise en compte non désirée de séparateurs qui donnent des cellules qui posent problème ?

Phase 3 [en amont] : Obtention des vidéos en ligne & choix des outils de sous-titrage

Plateforme Amara :

Traduction possible de vidéos déjà en ligne (Youtube, Dailymotion, Vimeo, celles de son propre site en formats Ogg, WebM, flv, mp4 …) notamment via l'upload de fichiers de sous-titres préexistants (formats acceptés : srt, .ssa, .sbv, .dfxp, .txt ou .xml pour dfxp).

1/Premie

Des outils de sous-titrage sous Linux, Subtitle Editor & Subtitle Composer, qui offrent l'un et l'autre une dizaine de formats de sortie différents.
L'incrustation en dur ultérieur dans la vidéo peut se faire au besoin avec VLC ou Avidemux.

Pas de problème réel rencontré lors de l'upload d'un fichier test .srt sur Amara (cf. mode d'emploi plus loin). Par contre, l'upload d'un .txt aboutit sur une non prise en compte des timelines préintégrées (à privilégier lorsqu'il n'y a qu'un script sans indications temporelles donc).
Ensuite, il a récupéré le code d'embarquement et l'a déposé sur un site avec des résultats convaincants.
Vu l'ergonomie et l'esprit de cet outil, on est donc partis sur cette option qui nous permet par ailleurs de découvrir une plateforme de sous-titrage collaborative de vidéos en ligne largement employée par des plateformes « prestigieuses » (Mozilla Foundation, conférences TED, Netflix, Udacity, etc).

2/Tests en mode « mettons nous en situation d'atelier »

S'est posée la question de la possibilité de travailler à plusieurs en même temps sur une même url source... ce n'est pas possible en l'état actuel des choses.

Le dernier qui a la main sur l'enregistrement des sous-titres modifiés en ligne a « raison » et c'est sa version qui est prise en compte. Pour permettre un flux de travail réellement collaboratif, synchronisé en temps réel et supervisé par un centralisateur, il est nécessaire de créer une équipe de traduction en contactant la plateforme, voire peut-être de payer ? A vérifier plus tard donc.

Pour l'atelier, on va devoir fonctionner en mode tournant et nous envisageons d'expliquer aux participants 3 type de manipulations de base dont les principes et méthodologies sont détaillés ci-dessous.

Plans B : synchronisation des sous-titres sur une vidéo « en dur »

1/Premiers tests par Gérald (sous Linux) :

Téléchargement de la vidéo avec cclive en format webm. L'inscrustation en dur s'est avérée assez laborieuse du fait que la vidéo captée de Youtube était encodée en VP8/Webm et que ce format n'est pas trop manipulable avec les outils à disposition. Un transcodage en h264 mkv a arrangé les choses.

Pour l'incrustation, Gérald testé différents outils : avconv (alternative à ffmpeg), avidemux et vlc. Aucun n'a produit un résultat sans faute dans un premier temps. Avconv a donné un résultat à peu près correct mais mangeait le premier sous-titre. Dans un deuxième temps, Gérald est parvenu à intégrer les sous-titres sans problème grâce à fmpeg.

Deux extraits des 3 premières minutes du documentaire ont été traités ici
http://www.fdn.fr/~gouvradou/spip.php?article25
http://www.fdn.fr/~gouvradou/spip.php?article26 (comme le média n'est pas libre, accès authentifié => login "subtest", mdp, "subtestmdl29" (cliquer en bas de la page "Se connecter").

2/Tests (sous Linux & Windows)

Obtention de la vidéo via http://www.telechargervideoyoutube.com
Formats de sortie possibles et exemples de résolutions/compressions

.webm	Moyenne - 360p	VP8	0.50 Mbit/s	Vorbis	128 kbit/s
.mp4	Moyenne - 360p	H.264	0.50 Mbit/s	AAC	96 kbit/s
.flv	Basse - 240p	Sorenson H.263	0.25 Mbit/s	MP3	64 kbit/s
.3gp	Basse - 240p	MPEG-4 Visual	0.18 Mbit/s	AAC	36 kbit/s
.3gp	Basse - 144p	MPEG-4 Visual	0.05 Mbit/s	AAC	24 kbit/s

Le format choisi a été le mp4 pour tenter un format différent du précédent et ça roule sur VLC (yeaaah :o)

Sous-titrage sur vidéo en dur grâce à une solution libriste recommandée par April : AegiSub On a obtenu AegiSub sur sa plateforme d'origine, en versions pour Windows et Linux. Le fonctionnement de base est assez simple.

Un tutoriel relativement complet est disponible à l'url suivante :
http://thehypnotizer.free.fr/tutoaegi/tut.htm

Des recommandations pratiques d'ordre général concernant la lisibilité et la visibilité des sous-titres sont par ailleurs compilées par un contributeur du blog Qelios :
http://blog.qelios.net/sous-titrage-videos-optimiser-aspect-pour-ameliorer-ux/

Phase 4 [atelier] : Synchronisation et mise en forme des sous-titres [Aegis], intégration et vérifications en ligne [Amara]

Il s'agirait de diviser l'atelier en plusieurs sous-ateliers tournants (en fonction du nombre de participants, découpage en tronçons de script traduit allant de 5min à 15min).

2 ATELIERS PRINCIPAUX :

Resynchronisation des sous-titres pour générer un fichier basique .srt (ou .txt) via AegiSub :

-Téléchargement et installation du logiciel AegiSub sur les PC (et portables?)
-Découverte et prise en main d'AegiSub (tutoriel et conseils)
-Imports des fichiers (vidéo, audio, textes)
-Premiers pas avec modifications des sous-titres (textes : réajustement des sauts de lignes, coupes de textes si morceaux de phrases trop longs, et),
-Resynchronisation des textes sur le son lorsque c'est nécessaire : simples modifications des bornes temporelles de sous-titres sur la base de l'analyse sonore ou redécoupage de la timeline existante (2 sous-titres > 3 sous-titres par exemple).
-Post-process (cf. tutoriel)
-Exports en .srt des tronçons au fur et à mesure
> Compilation intégrale d'un .srt unique à réaliser au fur et à mesure en mettant les fichiers bouts à bouts au cours de la journée) > Chaque tronçon pris en charge par un participant devrait de préférence être versé au fichier commun une fois achevé.

Vérification des sous-titres et uploads sur vidéos en ligne via Amara

-Découverte et prise en main de la plateforme, inscriptions (accès privés, comptes factices ?)
-Manips pour imports des fichiers .srt issus de l'étape précédente (écrasement des sous-titres uploadés les uns après les autres)
-Visionnages et vérifications de la fluidité (+ coquilles éventuelles), modifications éventuelles
-Manips d'exports des versions vérifiées et corrigées
-Compilation intégrale d'un .srt unique à réaliser au fur et à mesure en mettant les fichiers bouts à bouts au cours de la journée > Chaque tronçon pris en charge par un participant devrait de préférence être versé au fichier commun une fois achevé.

2 ATELIERS COMPLEMENTAIRES :

Mise en forme plus élaborée des sous-titres sur AegiSub :

-Découverte et prise en main de l'outil (tutoriel).
-Imports des fichiers (vidéo, audio, textes)
-Modifications des sous-titres via balises de codes : gras, italiques, couleurs
-Incrustation de nouveaux sous-titres positionnés pour éléments hors texte (séquences films)
-Génération d'un fichier .ass

Découverte des manips de saisie de sous-titres en mode « sans rien » sur Amara :

-Découverte et prise en main de la plateforme, inscriptions (accès privés, comptes factices ?)
-Saisie des sous-titres
-Manips de synchronisation
-Enregistrement & export d'un fichier de sous-titres
-Embarquement de la vidéo sous-titrée sur site perso (par exemple).